

Iterative scheme for solving optimal transportation problems arising in reflector design

Tilmann Glimm and Nick Henscheid

Department of Mathematics

Western Washington University, Bellingham, WA 98225

Abstract

We consider the geometrical optics problem of finding a system of two reflectors that transform a spherical wavefront into a beam of parallel rays with prescribed intensity distribution. Using techniques from optimal transportation theory, it has been shown before that this problem is equivalent to an infinite dimensional linear programming (LP) problem. We investigate techniques for constructing the two reflectors numerically. A straightforward discretization of this problem has the disadvantage that the number of constraints increases rapidly with the mesh size. So with this technique only very coarse meshes are practical. To address this well-known issue we propose an iterative solution scheme. In each step an LP problem is solved. Information from the previous iteration step is used to reduce the number of constraints necessary. As a proof of concept we apply our proposed scheme to solve a problem with synthetic data. We give evidence that the scheme converges. We also show that it allows for much finer meshes than a simple discretization scheme. There exists a growing literature for the application of optimal transportation theory to other beam shaping problems. Our proposed scheme is easy to adapt for these problems as well.

1 Introduction

The following beam shaping problem from geometric optics was described in [15]; see Figure 1: Suppose a point source emits a spherical wavefront with a given intensity distribution. The problem at hand consists of transforming this input beam into an output beam of parallel light rays with a prescribed intensity distribution. This transformation is to be achieved with a system of two reflectors. The problem has some practical importance in engineering, see further literature cited in [15].

The first rigorous mathematical solution to the problem was provided in [10], using an approach based on the theory of optimal transportation [5, 6, 8, 19]. See also the references [11, 9, 21, 12, 16] which deal with other beam shaping problems using related techniques.

The result of [10] is summarized in section 2 below, with Theorem 2.5 stating the main result. The central feature is that the original reflector design problem is reformulated as an infinite dimensional constrained optimization problem, namely the problem of minimizing a certain functional on a function space. It is the dual problem for the problem of finding a map from the input aperture to the output aperture which minimizes a certain cost functional. (See Problem II in section 2 for the exact statement.)

This reformulation of the problem is not only of theoretical value for questions of existence and uniqueness of solutions, but it also translates into a practical method for finding the solution. In fact, the discretization of the infinite dimensional constrained optimization problem is a standard linear programming problem and can be solved numerically. In this paper, we describe a numerical scheme for solving these problems. An immediate obstacle is that the number of constraints becomes very large even for relatively coarse meshes – if there are M sample points in the input aperture (denoted by \bar{D} in Figure 1) and N points in the output aperture (\bar{T} in Figure 1), then the number of constraints is $M \cdot N$. We found that with a simple straightforward discretization method, standard LP solvers (on PCs with up to 4 GB RAM) could not handle more than approximately 500 sample points on the domains \bar{T} and \bar{D} – a number which is arguably too small for many applications. For this reason, we devised a more elaborate iterative method, where discretized systems are solved on finer and finer grids, in each step using information from the previous solution to choose only a subset of all possible constraints. This slows the growth of the number of constraints. (Details are described in Section 3.) With this step-wise mesh refinement scheme, as a proof of concept, we were able to obtain solutions on meshes with about 4500 points on each aperture using MATLAB and a standard LP solver. Using compiled computer language and more specialized solvers for optimal transportation problems, we expect that this number could still be substantially improved.

There has been increased interest in numerical methods for optimal transportation problems in the last 10 years. Most work has concentrated on the Monge-Ampère equation, which arises in the special case of optimal transport in \mathbb{R}^n with quadratic cost (also called L^2 optimal transport) [14, 2, 3, 7, 13, 23], although some authors have treated costs proportional to the distance (L^1 optimal transport) [1]. These methods are based on fluid mechanical approaches [2] or various finite difference approaches [3]. They are generally faster and allow for much larger mesh sizes than methods based on a discretization of the linear programming problem, but since they use the special structure of the Monge-Ampère equation, they are not directly applicable to more general cost functions or more general manifolds.

One of the few papers that numerically addressed more general situations for optimal transport is [18]. A similar approach to the solution was taken there as we do in this paper, namely a discretization of the linear programming problem. (The authors in [18] used the primal formulation of the optimal transportation problem, whereas our approach is equivalent to using the dual formulation.) The authors noted that the number of variables is growing very fast with mesh sizes, making it impossible to solve the problem numerically even for moderate mesh sizes due to memory limitations. In fact, they noted that the software they used, the package Soplex [22], was designed to handle up to 2 million variables. This corresponds to mesh sizes of approximately 700 points on the input and output aperture in our problem if one employs a straightforward discretization scheme. The authors noted that for better results, one needs carefully designed programs. This is what we supply here for our problem: As explained above, our “mesh refinement” approach addresses this problem and makes it feasible to solve the problem for finer meshes.

We note that our proposed algorithm does not make any *a priori* symmetry assumptions on the form the reflectors. It can also very easily be adapted for the numerical solution of other beam shaping problems for which a formulation using optimal transportation theory have been found, for example those in [9, 11, 20].

This article is organized as follows: In section 2, we recall the reformulation of the reflector construction problem as a linear programming transportation as given in [10] and fix some notation. We then describe a basic discretization scheme for the numerical solution and propose

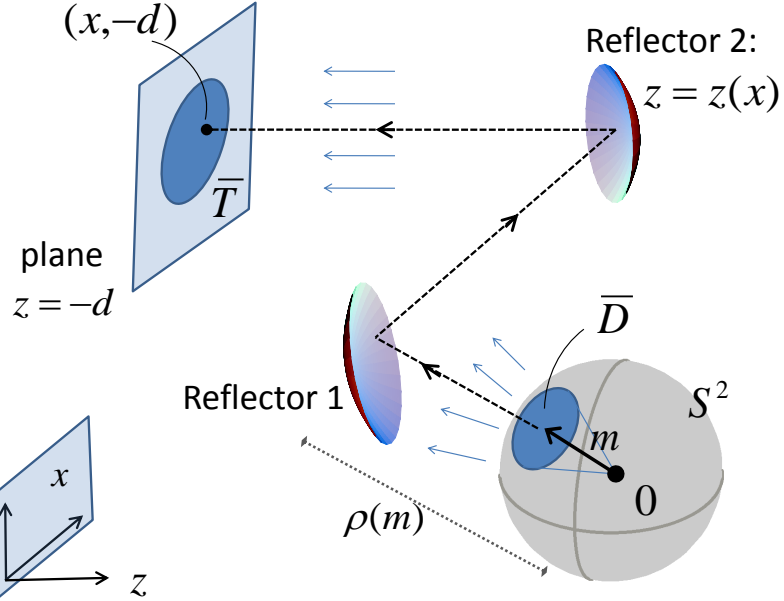


Figure 1: Sketch of the reflector problem. The point source is located at the origin $\mathbf{0}$ of the coordinate system. Note the conventions on coordinates as illustrated by the coordinate system in the lower left hand corner: The output beam propagates in the direction of the *negative* z -axis, and points in the plane perpendicular to the z -axis are denoted by the vector $\mathbf{x} \in \mathbb{R}^2$. Thus points in three dimensional space are denoted by (\mathbf{x}, z) . See Section 2 for more. (Adapted from [10].)

an improved “mesh refinement” scheme in section 3. The following section 4 is devoted to numerical tests of this scheme. We first derive an explicit analytical solution to be used as synthetic data for our numerical work. Then we compute the solution numerically and analyze the error of approximation. Our scheme requires a sequence $\varepsilon^{(1)}, \varepsilon^{(2)}, \dots$ of “constraint inclusion thresholds.” We analyze different choices of this sequence. We conclude with a brief summary and propose future work.

2 Formulation as Linear Programming Problem

We now briefly review the notation for the problem posed in [15], as well as the result of [10], which reformulates the reflector design problem as a linear programming problem.

Consider the configuration show in Figure 1. A point source at the origin $O = (0, 0, 0)$ generates a spherical wave front over a given input aperture \bar{D} contained in the unit sphere S^2 . The input beam has a given intensity distribution. By means of two reflectors, this wave front is to be transformed into a beam of parallel rays propagating in the direction of the negative z -axis. This output beam has to have a prescribed intensity distribution. The cross section of the output beam in a plane perpendicular to the direction of propagation is called the output aperture, and denoted by \bar{T} . (Certain regularity conditions apply to \bar{D} and \bar{T} , see [10] for the

technical details.)

Denote points in space \mathbb{R}^3 by pairs (\mathbf{x}, z) , where $\mathbf{x} \in \mathbb{R}^2$ is the position vector in a plane perpendicular to the direction of propagation and $z \in \mathbb{R}$ is the coordinate in the (negative) direction of propagation. See again Figure 1 for our convention on the direction of the z -axis. Points on the unit sphere S^2 will typically be denoted by $\mathbf{m} \in S^2$; their components are also written as $\mathbf{m} = (\mathbf{m}_x, m_z)$ with $|\mathbf{m}_x|^2 + m_z^2 = 1$.

We fix the output aperture in the plane $z = -d$. We will seek to represent the first reflector as the graph of its polar radius $\rho(\mathbf{m})$ (for $\mathbf{m} \in \bar{D}$), and the second reflector as the graph of a function $z(\mathbf{x})$ (for $\mathbf{x} \in \bar{T}$). (See Figure 1.) That is

$$\text{Reflector 1: } R_1 = \{\rho(\mathbf{m}) \cdot \mathbf{m} \mid \mathbf{m} \in \bar{D}\}, \quad \text{Reflector 2: } R_2 = \{(\mathbf{x}, z(\mathbf{x})) \mid \mathbf{x} \in \bar{T}\}. \quad (1)$$

The geometrical optics approximation is assumed. It follows from general principles of geometric optics that all rays will have equal length from $(0, 0, 0)$ to the plane $z = -d$; this length is called the optical path length and will be denoted by L . We define the *reduced optical path length* as $\ell = L - d$.

Oliker [15] showed that local energy conservation translates into a complicated partial differential equation of Monge-Ampère type for $\rho(\mathbf{m})$. As noted in [15], the resulting equation is quite involved and a rigorous analysis of this equation seems very difficult. (See equation (59) in [15].)

To amend this, the problem was reformulated in [10] as a linear programming problem, which makes a complete analysis possible, both concerning theoretical results on existence and uniqueness, and gives a method for practical computations.

For this, the following function $K(\mathbf{m}, \mathbf{x})$, called the cost function in analogy with the theory of optimal transportation, plays an important role:

$$K(\mathbf{m}, \mathbf{x}) = \frac{\ell - \langle \mathbf{m}_x, \mathbf{x} \rangle}{2\ell(\ell^2 - |\mathbf{x}|^2)(1 + m_z)} - \frac{1}{4\ell^2} \quad \text{for } \mathbf{m} = (\mathbf{m}_x, m_z) \in \bar{D}, \mathbf{x} \in \bar{T}.$$

In further preparation, the following two transformations are needed:

Definition 2.1. Let $z = z(\mathbf{x})$ be a continuous function defined on $\bar{T} \subseteq \mathbb{R}^2$. Then define the function

$$\tilde{z}(\mathbf{x}) = \frac{1}{2\ell} - \frac{z(\mathbf{x})}{\ell^2 - |\mathbf{x}|^2} \quad \text{for } \mathbf{x} \in \bar{T}. \quad (2)$$

Definition 2.2. Let $\rho = \rho(\mathbf{m})$ be a continuous function defined on $\bar{D} \subseteq S^2$ with $\rho > 0$. Then define the function

$$\hat{\rho}(\mathbf{m}) = -\frac{1}{2\ell} + \frac{1}{2\rho(\mathbf{m}) \cdot (m_z + 1)} \quad \text{for } \mathbf{m} \in \bar{D}. \quad (3)$$

With these preparations, in [10], the following notion of a quasi-reflector pair and its associated ray tracing map is used. (The term used in [10] is “reflector pair”, but we use “quasi-reflector pair” here for clarity.)

Definition 2.3. A pair $(\rho, z) \in C_{>0}(\bar{D}) \times C(\bar{T})$ is called a *quasi-reflector pair* if $\hat{\rho}, \tilde{z} > 0$ and

$$\hat{\rho}(\mathbf{m}) = \sup_{\mathbf{x} \in \bar{T}} \left(\frac{1}{\tilde{z}(\mathbf{x})} K(\mathbf{m}, \mathbf{x}) \right) \quad \text{for } \mathbf{m} \in \bar{D}, \quad (4)$$

$$\tilde{z}(\mathbf{x}) = \sup_{\mathbf{m} \in \bar{D}} \left(\frac{1}{\hat{\rho}(\mathbf{m})} K(\mathbf{m}, \mathbf{x}) \right) \quad \text{for } \mathbf{x} \in \bar{T}. \quad (5)$$

Definition 2.4. Let $(\rho, z) \in C_{>0}(\bar{D}) \times C(\bar{T})$ be a quasi-reflector pair. Define its *reflector map*, or *ray tracing map*, as a set-valued map $\gamma: \bar{D} \rightarrow \{\text{subsets of } \bar{T}\}$ via

$$\gamma(\mathbf{m}) = \{\mathbf{x} \in \bar{T} \mid \hat{\rho}(\mathbf{m}) = \frac{1}{\tilde{z}(\mathbf{x})} K(\mathbf{m}, \mathbf{x})\} \quad \text{for } \mathbf{m} \in \bar{D}.$$

In [10], it is shown that $\gamma(\mathbf{m})$ is in fact single-valued for almost all $\mathbf{m} \in \bar{D}$. Therefore γ may be regarded as a transformation $\gamma: \bar{D} \rightarrow \bar{T}$.

If (ρ, z) is a “quasi-reflector pair” in the above sense, the following can be shown [10], justifying the choice of nomenclature: If physical copies of the corresponding surfaces (1) are made from a reflective material, then a ray emitted from the origin in the direction $\mathbf{m} \in \bar{D}$ will be reflected to a ray in the negative z -direction labeled by $\mathbf{x} = \gamma(\mathbf{m}) \in \bar{T}$.

With this, the reflector problem can be formulated rigorously:

Problem I. (Reflector Problem) For given input and output intensities $I(\mathbf{m}), \mathbf{m} \in \bar{D}$, and $L(\mathbf{x}), \mathbf{x} \in \bar{T}$, respectively, satisfying global energy conservation $\int_{\bar{D}} I(\mathbf{m}) d\sigma = \int_{\bar{T}} L(\mathbf{x}) dx$, find a pair $(\rho, z) \in C_{>0}(\bar{D}) \times C(\bar{T})$ that satisfies the following conditions:

- (i) (ρ, z) is a quasi-reflector pair in the sense of Definition 2.3.
- (ii) The ray tracing map $\gamma: \bar{D} \rightarrow \bar{T}$ satisfies

$$\int_{\gamma^{-1}(\tau)} I(\mathbf{m}) d\sigma = \int_{\tau} L(\mathbf{x}) dx$$

for any Borel set $\tau \subseteq \bar{T}$.

Here $d\sigma$ denotes the standard area element on the sphere S^2 . The second condition is local energy conservation.

As we indicate below, the Reflector Problem can be reformulated in the following form:

Problem II. Minimize the functional

$$\mathcal{F}(r, \zeta) = \int_{\bar{D}} r(\mathbf{m}) I(\mathbf{m}) d\sigma + \int_{\bar{T}} \zeta(\mathbf{x}) L(\mathbf{x}) dx$$

on the set $\text{Adm}(\bar{D}, \bar{T}) = \{(r, \zeta) \in C(\bar{D}) \times C(\bar{T}) \mid r(\mathbf{m}) + \zeta(\mathbf{x}) \geq \log K(\mathbf{m}, \mathbf{x}) \text{ for all } \mathbf{m} \in \bar{D}, \mathbf{x} \in \bar{T}\}$.

Problem II is the dual problem to the problem of finding a map P from $\bar{D} \subseteq S^2$ to $\bar{T} \subseteq \mathbb{R}^2$ which minimizes the transportation functional $P \mapsto \int_{\bar{D}} K(\mathbf{m}, P(\mathbf{m})) d\sigma$. Problem I and Problem II are equivalent, as expressed in the following theorem.

Theorem 2.5. [10] Let $(\rho, z) \in C_{>0}(\bar{D}) \times C(\bar{T})$ be a quasi-reflector pair. Then $(\log \hat{\rho}, \log \tilde{z}) \in \text{Adm}(\bar{D}, \bar{T})$. The following statements are equivalent:

- (i) (ρ, z) solves the Reflector Problem I.
- (ii) $(\log \hat{\rho}, \log \tilde{z})$ solves Problem II.

Thus solving the reflector construction problem in Problem I is equivalent to solving the infinite dimensional LP problem in Problem II. Indeed, it can be shown that a solution as in Theorem 2.5 exists. (See Corollary 6.5 in [10].) In the remainder of this paper, we will concentrate on numerical solutions for this problem.

3 Numerical Schemes

In the following, we describe two scheme for solving Problem II numerically. First, let us list explicitly the given data on which the solution depends:

- (i) domains¹ $\bar{D} \subseteq S^2$, $\bar{T} \subseteq \mathbb{R}^2$
- (ii) nonnegative integrable functions $I(\mathbf{m})$, $\mathbf{m} \in \bar{D}$, and $L(\mathbf{x})$, $\mathbf{x} \in \bar{T}$, with $\int_{\bar{D}} I d\sigma = \int_{\bar{T}} L dx$
- (iii) a reduced optical length ℓ
- (iv) the value $\rho(\mathbf{m}_1) = \rho_1$ for some fixed $\mathbf{m}_1 \in \bar{D}$, where ρ is the radial function of the first reflector.²

The first numerical scheme, Scheme 3.1 is a very straightforward discretization of Problem II via meshes on the domains \bar{D} and \bar{T} . Because of its simplicity and straightforwardness, we can informally call this a “brute force” technique. The problem with this scheme is that each pair of points from \bar{D} and \bar{T} gives rise to a constraint, and so the technique requires a lot of memory, so that only coarse meshes can be used. (This problem is very well known, see for example [18].) To address this issue, we propose a more sophisticated scheme, Scheme 3.2, which is an iterative scheme using finer and finer meshes, where in each iteration, information from the previous solution is utilized to reduce the number of constraints.

The technique depends on choosing a parameter ε , the constraint threshold, in each iteration. We then discuss the problem of choosing an appropriate sequence of such ε .

3.1 “Simple Discretization” Scheme

The characterization of solutions by Theorem 2.5 immediately gives a practical solution scheme for solving the Reflector Problem I, namely by a straightforward discretization of the equivalent Problem II. (The same algorithm was used in [18].)

Numerical Scheme 3.1. (“Simple Discretization”)

- Create a mesh in the input domain \bar{D} by choosing sets $d_1^{(0)}, d_2^{(0)}, \dots, d_{M^{(0)}}^{(0)} \subseteq \bar{D}$, where the interiors of any $d_i^{(0)}$ and $d_j^{(0)}$ are disjoint (for $i \neq j$), and $\bar{D} = \bigcup_{i=1}^{M^{(0)}} d_i^{(0)}$. (For instance, $d_1^{(0)}, d_2^{(0)}, \dots, d_{M^{(0)}}^{(0)}$ may be a triangulation of \bar{D} .)
- Similarly, create a mesh in the output domain \bar{T} by choosing sets $t_1^{(0)}, t_2^{(0)}, \dots, t_{N^{(0)}}^{(0)} \subseteq \bar{T}$, where the interiors of any $t_i^{(0)}$ and $t_j^{(0)}$ are disjoint (for $i \neq j$), and $\bar{T} = \bigcup_{j=1}^{N^{(0)}} t_j^{(0)}$.
- Choose sample points $\mathbf{m}_i^{(0)} \in d_i^{(0)}$ (for $i = 1, \dots, M^{(0)}$) and $\mathbf{x}_j^{(0)} \in t_j^{(0)}$ (for $j = 1, \dots, N^{(0)}$). Here we may assume that $\mathbf{m}_1^{(0)} = \mathbf{m}_1$ is the same point as given in the data in (iv) above.

¹The rigorous result in [10] made the additional technical assumption $(0, 0, -1), (0, 0, 1) \notin \bar{D}$. For practical purposes, these assumptions can often be dropped.

²Without this constraint, the reflectors are not uniquely determined. This can be seen in the formulation of Problem II. Indeed, any transformation of the form $r \mapsto r + c$, $\zeta \mapsto \zeta - c$ for constant c will leave the objective functional unchanged.

- Find the solution $\left(\{r_i^{(0)}\}_{i=1}^{M^{(0)}}, \{\zeta_j^{(0)}\}_{j=1}^{N^{(0)}}\right)$ of the following LP problem:

$$\begin{aligned} & \text{Minimize } \sum_{i=1}^{M^{(0)}} r_i^{(0)} I\left(\mathbf{m}_i^{(0)}\right) \text{ area}\left(d_i^{(0)}\right) + \sum_{j=1}^{N^{(0)}} \zeta_j^{(0)} L\left(\mathbf{x}_j^{(0)}\right) \text{ area}\left(t_j^{(0)}\right) \\ & \text{subject to } r_1^{(0)} = \log \hat{\rho}_1 \text{ (where } \hat{\rho}_1 \text{ is as given in the data in (iv) above),} \\ & \text{and } r_i^{(0)} + \zeta_j^{(0)} \geq \log K\left(\mathbf{m}_i^{(0)}, \mathbf{x}_j^{(0)}\right) \text{ for } i = 1, \dots, M^{(0)}; j = 1, \dots, N^{(0)}. \end{aligned}$$

- Find the numbers $\rho_i^{(0)}, i = 1, \dots, M^{(0)}$ and $z_j^{(0)}, j = 1, \dots, N^{(0)}$ such that $r_i^{(0)} = \log \hat{\rho}_i^{(0)}$, and $\zeta_j^{(0)} = \log \hat{z}_j^{(0)}$. (This is straightforward by taking the inverse of the transformations given in Definitions 2.1 and 2.2.)

Then $\rho_i^{(0)}$ is an approximation for the true value $\rho\left(\mathbf{m}_i^{(0)}\right)$ of the radial function of the first reflector, evaluated at the sample point $\mathbf{m}_i^{(0)}$ for $i = 1, \dots, M^{(0)}$. Similarly, $z_j^{(0)}$ is an approximation for the true value $z\left(\mathbf{x}_j^{(0)}\right)$ of the function describing the second reflector, evaluated at the sample point $\mathbf{x}_j^{(0)}$ for $j = 1, \dots, N^{(0)}$. (We are using the superscript “(0)” here to distinguish this solution from additional iterative approximations we will obtain with the iterative scheme described below.)

It is important to point out that this discretization also yields a discretized version of the ray tracing map γ (see Definition 2.4). Namely, for each index $i = 1, \dots, M^{(0)}$, there is at least one corresponding index j^* , $1 \leq j^* \leq N^{(0)}$ where the constraint corresponding to the pair (i, j^*) is active, that is, where we have

$$r_i^{(0)} + \zeta_{j^*}^{(0)} = \log K\left(\mathbf{m}_i^{(0)}, \mathbf{x}_{j^*}^{(0)}\right).$$

This means that the point $\mathbf{m}_{j^*}^{(0)}$ is approximately the image of the point $\mathbf{x}_i^{(0)}$ under the ray tracing map γ .

3.2 “Iterative Refinement” Scheme

As noted in the introduction, one of the drawbacks of scheme 3.1 is that the constraint set in the linear programming problem in the penultimate step becomes large very fast with finer meshes, and the corresponding problem becomes too large to handle for standard LP solvers. We were not able to solve problems with more than very roughly 1000 mesh points on a standard PC with 4 GB RAM ($M^{(0)} \approx 500, N^{(0)} \approx 500$), which is arguably too small for many applications.

We addressed this problem by developing an iterative scheme. First the problem is solved for a mesh with relative few sample points (say $M^{(0)} = N^{(0)} = 250$). Then a finer mesh is chosen, and the previous solution is used to reduce the number of constraints.

Specifically, we have the following scheme, depending on a number $\varepsilon^{(1)} > 0$, which we call the “inclusion threshold,” to be explained in detail below.

Numerical Scheme 3.2. (“Iterative Refinement”)

- Use scheme 3.1 to find an initial solution $(\{r_i^{(0)}\}_{i=1}^{M^{(0)}}, \{\zeta_j^{(0)}\}_{j=1}^{N^{(0)}})$ for $M^{(0)}$ sample points on \bar{D} and $N^{(0)}$ sample points on \bar{T} , respectively.

- Create a mesh on \bar{D} by choosing sets $d_1^{(1)}, d_2^{(1)}, \dots, d_{M^{(1)}}^{(1)} \subseteq \bar{D}$, where the interiors of any $d_i^{(1)}$ and $d_j^{(1)}$ are disjoint (for $i \neq j$), and $\bar{D} = \bigcup_{i=1}^{M^{(1)}} d_i^{(1)}$. Here $M^{(1)}$ is chosen larger than $M^{(0)}$, meaning that we have a finer mesh.
- Similarly, create a mesh on \bar{T} by choosing sets $t_1^{(1)}, t_2^{(1)}, \dots, t_{N^{(1)}}^{(1)} \subseteq \bar{T}$, where the interiors of any $t_i^{(1)}$ and $t_j^{(1)}$ are disjoint (for $i \neq j$), and $\bar{T} = \bigcup_{j=1}^{N^{(1)}} t_j^{(1)}$.
- Choose sample points $\mathbf{m}_i^{(1)} \in d_i^{(1)}$ (for $i = 1, \dots, M^{(1)}$) and $\mathbf{x}_j^{(1)} \in t_j^{(1)}$ (for $j = 1, \dots, N^{(1)}$). Here we may assume that $\mathbf{m}_1^{(1)} = \mathbf{m}_1$ is the same point as given in the data in (iv) above.

- Interpolate the initial solution $\{r_i^{(0)}\}_{i=1}^{M^{(0)}}$ to find an approximation for the first reflector. That is, by some interpolation method, find a continuous function $r(\mathbf{m})$, $\mathbf{m} \in \bar{D}$, with

$$r(\mathbf{m}_i^{(0)}) = r_i^{(0)} \quad \text{for } i = 1, \dots, M^{(0)}.$$

- Similarly, interpolate the initial solution $\{\zeta_j^{(0)}\}_{j=1}^{N^{(0)}}$ to find an approximation for the second reflector. That is, by some interpolation method, find a continuous function $\zeta(\mathbf{x})$, $\mathbf{x} \in \bar{T}$, with

$$\zeta(\mathbf{x}_j^{(0)}) = \zeta_j^{(0)} \quad \text{for } j = 1, \dots, N^{(0)}.$$

- Find the solution $(\{r_i^{(1)}\}_{i=1}^{M^{(1)}}, \{\zeta_j^{(1)}\}_{j=1}^{N^{(1)}})$ of the following LP problem:

$$\text{Minimize } \sum_{i=1}^{M^{(1)}} r_i^{(1)} I(\mathbf{m}_i^{(1)}) \text{ area}(d_i^{(1)}) + \sum_{j=1}^{N^{(1)}} \zeta_j^{(1)} L(\mathbf{x}_j^{(1)}) \text{ area}(t_j^{(1)})$$

$$\text{subject to } r_1^{(1)} = \log \hat{\rho}_1,$$

$$\text{and } r_i^{(1)} + \zeta_j^{(1)} \geq \log K(\mathbf{m}_i^{(1)}, \mathbf{x}_j^{(1)})$$

$$\text{for those pairs } (i, j) \text{ with } r(\mathbf{m}_j^{(1)}) + \zeta(\mathbf{x}_j^{(1)}) - \log K(\mathbf{m}_i^{(1)}, \mathbf{x}_j^{(1)}) < \varepsilon^{(1)}.$$

- Find the numbers $\rho_i^{(1)}$, $i = 1, \dots, M^{(1)}$, and $z_j^{(1)}$, $j = 1, \dots, N^{(1)}$, such that $r_i^{(1)} = \log \hat{\rho}_i^{(1)}$, and $\zeta_j^{(1)} = \log \hat{z}_j^{(1)}$.

The idea of the second scheme 3.2 is to solve the discretized LP on a coarse mesh first and then use this information to reduce the number of constraints needed for the LP on a finer mesh. To wit, the difference between schemes 3.1 and 3.2 is that in the discretized LP problem, in scheme 3.2 not *all* pairs of sample points from \bar{D} and \bar{T} , represented by pairs of indices (i, j) , are included in the list of constraints. In fact, we would in principle only need to include those where the constraint is active, that is, where the corresponding inequality holds with equality. Of course, this information is not available *a priori*. Instead, we use the following heuristic: A constraint should only be included in the LP problem if it is “almost” active when we use an

interpolation of the solution on the coarse mesh as an approximate solution on the fine mesh. This is where the “inclusion threshold” $\varepsilon^{(1)}$ come into play. Namely, we only include those constraints (i, j) for which the difference $r(\mathbf{m}_j^{(1)}) + \zeta(\mathbf{x}_j^{(1)}) - \log K(\mathbf{m}_i^{(1)}, \mathbf{x}_j^{(1)})$ is less than the inclusion threshold $\varepsilon^{(1)}$. (Here $r(\mathbf{m})$ and $\zeta(\mathbf{x})$ are interpolations of the solution of the LP problem for the coarse mesh.) Clearly, increasing $\varepsilon^{(1)}$ means more constraints are included, but also potentially the solution of the LP represents a better approximation of the exact reflector pair.

In our numerical tests, we found it advantageous to scale the functions $I(\mathbf{m})$ and $L(\mathbf{x})$ so that the approximations for the integrals $\int_{\bar{D}} I d\sigma$ and $\int_T b L dx$ yield exactly the same result.

Note that the algorithm in scheme 3.2 can be iterated again. We can use the first iterative solution $\rho_i^{(1)}, i = 1, \dots, M^{(1)}$, and $z_j^{(1)}, j = 1, \dots, N^{(1)}$, in Scheme 3.2 to obtain a second iterative solution $\rho_i^{(2)}, i = 1, \dots, M^{(2)}$, and $z_j^{(2)}, j = 1, \dots, N^{(2)}$, and so on.

3.3 Choice of thresholds for Iterative Refinement scheme 3.2

For the iteration of scheme 3.2 described above, we have to choose a corresponding sequence of mesh sizes $\{(M^{(0)}, N^{(0)}), (M^{(1)}, N^{(1)}), (M^{(2)}, N^{(2)}), \dots\}$. (Here $M^{(k)}$ denotes the number of sample points for the input aperture \bar{D} , and $N^{(k)}$ denotes the number of sample points for the target aperture \bar{T} , both in the k^{th} iteration of the scheme.) We also need to choose a corresponding sequence of thresholds $\varepsilon^{(1)}, \varepsilon^{(2)}, \varepsilon^{(3)}, \dots$.

Let us assume that the sequence $\{(M^{(0)}, N^{(0)}), (M^{(1)}, N^{(1)}), (M^{(2)}, N^{(2)}), \dots\}$ is given. This raises the question: How should the corresponding sequence of thresholds $\varepsilon^{(1)}, \varepsilon^{(2)}, \varepsilon^{(3)}, \dots$ be picked?

This is a question of great practical importance. To wit, if the sequence $\{\varepsilon^{(k)}\}_k$ is constant or decreases too slowly, then “many” constraints are included in each iteration, meaning that the LP problems will become large fast and we will run out of memory quickly after a few iterations. If the sequence $\{\varepsilon^{(k)}\}_k$ decreases too fast, then “few” constraints are included in each iteration. This could cause for instance that some index i (with $1 \leq i \leq M^{(k)}$) may not even be included in any of the constraints with any of the indices j (with $1 \leq j \leq N^{(k)}$), or vice versa. This would cause the LP problem to be unbounded and hence there would be no solution. (This behavior is illustrated in the practical tests summarized in Table 1; see also the discussion in section 4.)

We can make a very rough estimate for a good choice of the sequence $\{\varepsilon^{(k)}\}_k$, assuming for simplicity that $M^{(k)} \approx N^{(k)}$. Let us assume that each sample point $m_i^{(k)}$ in the input aperture D is paired via the ray tracing map γ with a unique point $\mathbf{x}_{j(i)}^{(k)} \approx \gamma(m_i^{(k)})$ and vice versa. Thus there are approximately $M^{(k)}$ such pairs. Let $r(\mathbf{m}), \mathbf{m} \in \bar{D}$, and $\zeta(\mathbf{x}), \mathbf{x} \in \bar{T}$, denote the function obtained by interpolating the solution of the $(k-1)^{th}$ iterate, $\left(\{r_i^{(k-1)}\}_{i=1}^{M^{(k-1)}}, \{\zeta_j^{(k-1)}\}_{j=1}^{N^{(k-1)}}\right)$. Then the set of all points in (\mathbf{m}, \mathbf{x}) -space that satisfy $r(\mathbf{x}) - z(\mathbf{x}) - \log K(\mathbf{m}, \mathbf{x}) < \varepsilon^{(k)}$ is approximately an ellipsoid for small $\varepsilon^{(k)}$. This can be seen by using the Taylor expansion around the pair $(\mathbf{m}_i^{(k)}, \mathbf{x}_{j(i)}^{(k)})$ in (\mathbf{m}, \mathbf{x}) -space. The volume of this ellipsoid is proportional to $(\varepsilon^{(k)})^2$. Since each pair $(\mathbf{m}_i^{(k)}, \mathbf{x}_{j(i)}^{(k)})$ that satisfies this inequality corresponds to a constraint included in the LP of the k^{th} iteration, the total number of constraints should be roughly proportional to $M^{(k)} \cdot (\varepsilon^{(k)})^2$. So to keep the number of constraints

approximately constant throughout the iterations, based on these heuristics, one may choose

$$\varepsilon^{(k)} = \frac{C}{\sqrt{M^{(k)}}},$$

where C is a constant. These heuristics are very rough and in practice the number of constraints is still increasing with $M^{(k)}$, albeit at a much slower rate than the case of constant $\varepsilon^{(k)}$. In section 4.3, we describe an implementation of this idea.

4 Numerical Tests

To test the validity of the numerical scheme described above, we used it on a case where the solution is known in analytic form. In the next subsection, we first describe this special analytic solution, and then discuss our results.

4.1 An Analytic Solution

In order to obtain a data set where an explicit analytic solution is known, we consider a special given configuration of reflectors, and then solve the “forward” problem of determining the output intensity this system produces for a given input intensity.

4.1.1 Construction of a pair of reflectors R_1, R_2

Consider the following set of two reflectors R_1 and R_2 , sketched in Figure 2: Let $\mathbf{a} = (a, b, c)$ be a given point in \mathbb{R}^3 , and let $R > 0$ and $\alpha > 0$ be two positive numbers with $R > |\mathbf{a}| = \sqrt{a^2 + b^2 + c^2}$. Now let the first reflector R_1 be the boundary of a spheroid with foci at the origin \mathbf{O} and at \mathbf{a} and with major diameter $R/2$. (Thus for each point \mathbf{P} on R_1 , the sum of the distances from each of the two foci $\overline{\mathbf{PO}} + \overline{\mathbf{Pa}}$ equals R .) Let the second reflector R_2 be the boundary of a paraboloid whose main axis is the negative z -axis and whose focus is at \mathbf{a} with focal parameter 2α . (Thus for each point $\mathbf{p} = (x, y, z)$ on R_2 , the sum of the distance to the focus and the shifted z -component $\overline{\mathbf{Pa}} + (z - c)$ equals 2α .)

Note that by definition of the two reflectors, any cone of light rays emitted from the origin \mathbf{O} will be transformed into a beam of parallel rays traveling in the direction of the negative z -axis. Indeed, a ray emitted from \mathbf{O} will be reflected off R_1 towards the focus \mathbf{a} . This ray will then be reflected in the direction of the negative z -axis by R_2 . See again Figure 2.

It is not hard to find explicit expressions for the two reflectors. If we write $R_1 = \{\rho(\mathbf{m}) \cdot \mathbf{m} \mid \mathbf{m} \in S^2\}$ and $R_2 = \{(\mathbf{x}, z(\mathbf{x})) \mid \mathbf{x} = (x, y) \in \mathbb{R}^2\}$, then we have the following expressions:

$$\text{radial function } \rho \text{ for } R_1: \rho(\mathbf{m}) = \frac{R^2 - |\mathbf{a}|^2}{2(R - \mathbf{m} \cdot \mathbf{a})} \quad (6)$$

and

$$\text{equation for } R_2: -4\alpha q = |\mathbf{p}|^2 - 4\alpha^2 \quad (7)$$

where we used the shifted coordinates $(\mathbf{p}, q) = (x, y, z) - (a, b, c) = (\mathbf{x}, z) - \mathbf{a}$.

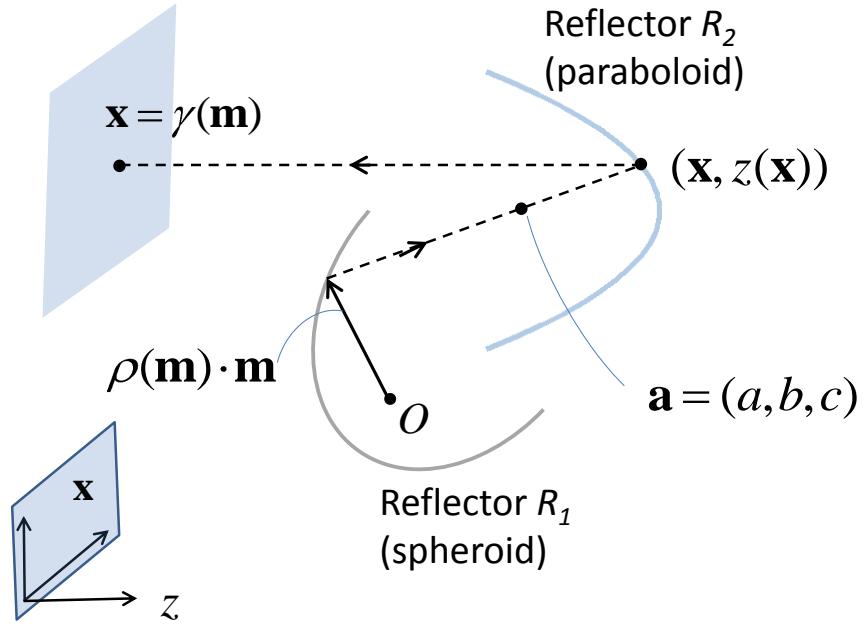


Figure 2: Sketch of the reflectors R_1 and R_2 . R_1 is the boundary of an ellipsoid whose foci are the points O and \mathbf{a} . R_2 is the boundary of a paraboloid whose focus is \mathbf{a} . The sketch shows a two-dimensional cross section. A ray given by the direction $\mathbf{m} \in S^2$ will be reflected by R_1 and then by R_2 to a ray in the direction of the negative z -axis.

4.1.2 Ray Tracing Map γ

One can now determine the ray tracing map γ corresponding to the reflector pair R_1, R_2 . See again Figure 2.

Let $\mathbf{m} \in S^2$ be a given direction. Thus $\mathbf{m} = (\mathbf{m}_x, m_z)$ with $\mathbf{m}_x \in \mathbb{R}^2$ and $|\mathbf{m}_x|^2 + m_z^2 = 1$. To determine the image $\gamma(\mathbf{m})$, consider the ray emitted from the origin in the direction \mathbf{m} . The ray will encounter the first reflector at the point $\rho(\mathbf{m}) \cdot \mathbf{m}$ and will then be reflected towards the second focus \mathbf{a} . Thus the reflected ray can be parameterized by $\mathbf{a} + \lambda \cdot (\mathbf{a} - \rho(\mathbf{m}) \cdot \mathbf{m})$, where λ is a parameter. With equation (7), this yields that the point where the reflected ray hits the second reflector is $\mathbf{a} + \lambda^* \cdot (\mathbf{a} - \rho \cdot \mathbf{m})$ where

$$\lambda^* = \frac{2\alpha}{(R - \rho)^2 - (c - \rho m_z)^2} ((R - \rho) - (c - \rho m_z)) = \frac{2\alpha}{R + c - \rho(1 + m_z)}.$$

The projection of this point to the xy -plane is the value of the ray tracing map. Thus we have the following explicit formula for the ray tracing map:

$$\gamma(\mathbf{m}) = \begin{pmatrix} a \\ b \end{pmatrix} + \frac{2\alpha}{R + c - \rho(\mathbf{m})(1 + m_z)} \left(\begin{pmatrix} a \\ b \end{pmatrix} - \rho(\mathbf{m}) \mathbf{m}_x \right), \quad \mathbf{m} = (\mathbf{m}_x, m_z) \in S^2. \quad (8)$$

4.1.3 Solution of the Forward Problem

We can now solve the forward problem: Given the reflector pair R_1, R_2 as defined above, an input aperture $\bar{\Omega} \subseteq S^2$ and an intensity distribution $I(\mathbf{m})$, $\mathbf{m} \in \bar{\Omega}$, find the output aperture $\bar{T} \subseteq \mathbb{R}^2$ and the output intensity $L(\mathbf{x})$, $\mathbf{x} \in \bar{T}$.

The output aperture is simply the image of $\bar{\Omega}$ under the ray tracing map γ : $\bar{T} = \gamma(\bar{\Omega})$. To find the induces intensity $L(\mathbf{x})$, use the defining property

$$\int_{\omega} I(\mathbf{m}) d\sigma = \int_{\gamma(\omega)} L(\mathbf{x}) d\mathbf{x} \quad (9)$$

for all Borel sets $\omega \subseteq \bar{\Omega}$. This is an energy balance equation.

The above integral equation allows us to find an explicit expression for $L(\mathbf{x})$ given $I(\mathbf{m})$, or vice versa. We consider for simplicity the case that $\bar{\Omega}$ is contained in the left hemisphere $S_-^2 = S^2 \cap \{(x, y, z) \in \mathbb{R}^3 \mid z < 0\}$. We can then use coordinates

$$\tau: \{(m_x, m_y) \in \mathbb{R}^2 \mid m_x^2 + m_y^2 < 1\} \rightarrow S_-^2, \quad (10)$$

$$(x, y) \mapsto \left(m_x, m_y, -\sqrt{1 - m_x^2 - m_y^2} \right). \quad (11)$$

In these coordinates, the standard measure on S^2 is given by $d\sigma = \frac{dm_x dm_y}{|m_z|}$, where

$m_z = -\sqrt{1 - m_x^2 - m_y^2}$. Using this, (9) is then equivalent to the equation

$$I(\mathbf{m}_x, m_z) \frac{1}{|m_z|} = L(\gamma(\tau(\mathbf{m}_x))) J(\gamma \circ \tau). \quad (12)$$

Here $J(\gamma \circ \tau) = \left| \det \begin{pmatrix} \frac{\partial(\gamma \circ \tau)}{\partial m_x} & \frac{\partial(\gamma \circ \tau)}{\partial m_y} \end{pmatrix} \right|$ denotes the Jacobian of the map $\gamma \circ \tau$, With the help

of a computer algebra system like Mathematica, this can be evaluated explicitly as

$$J(\gamma \circ \tau) = \frac{4\alpha^2(|\mathbf{a}|^2 - R^2)^2}{-m_z \left(2(c+R) \mathbf{m}_{\mathbf{x}} \cdot \begin{pmatrix} a \\ b \end{pmatrix} - (1+m_z)|a,b|^2 - (c+R)^2(1-m_z) \right)^2}, \quad (13)$$

where again $m_z = -\sqrt{1 - m_x^2 - m_y^2}$.

4.2 A data set

We can now construct a particular data set using the results of the previous sections. We pick $a = b = 0$, and $c = -0.4$, $\alpha = 1$, $R = 1.3$, and consider the input aperture

$$\bar{\Omega} = \{(m_x, m_y, m_z) \in S^2 \mid \sqrt{m_x^2 + m_y^2} \leq 0.8, m_z < 0\},$$

which is a spherical cap centered at the point $(0, 0, -1)$ ³. Using the results from the previous subsections, we can now find the output aperture in a straightforward manner:

$$\bar{T} = \{(x, y) \in \mathbb{R}^2 \mid \sqrt{x^2 + y^2} \leq 1.8888888889\}.$$

We choose a constant output intensity

$$L(\mathbf{x}) = 1 \quad \text{for } \mathbf{x} \in \bar{T}.$$

Using the relation (12), this gives the input intensity

$$I(\mathbf{m}) = \frac{14.2716049383}{(1 - m_z)^2} \quad \text{for } \mathbf{m} = (m_x, m_y, m_z) \in \bar{\Omega}.$$

The two reflectors are given as follows:

$$\text{radial radius } \rho \text{ for } R_1: \rho(\mathbf{m}) = \frac{0.765}{1.3 + 0.4 m_z} \quad (14)$$

and

$$\text{equation for } R_2: z = -0.25 (x^2 + y^2) + 0.6. \quad (15)$$

The corresponding reduced optical path length is

$$\ell = R + 2\alpha = 2.9.$$

This follows from the construction of the two reflectors and the properties of ellipsoids and paraboloids. The two reflectors are shown in Figure 3.

³The data do not represent a physically possible set of reflectors since there would be blockage. However, the example is useful as a proof of concept, and the numerical scheme itself is completely independent of the shape of the apertures or any *a priori* symmetries of the problem.

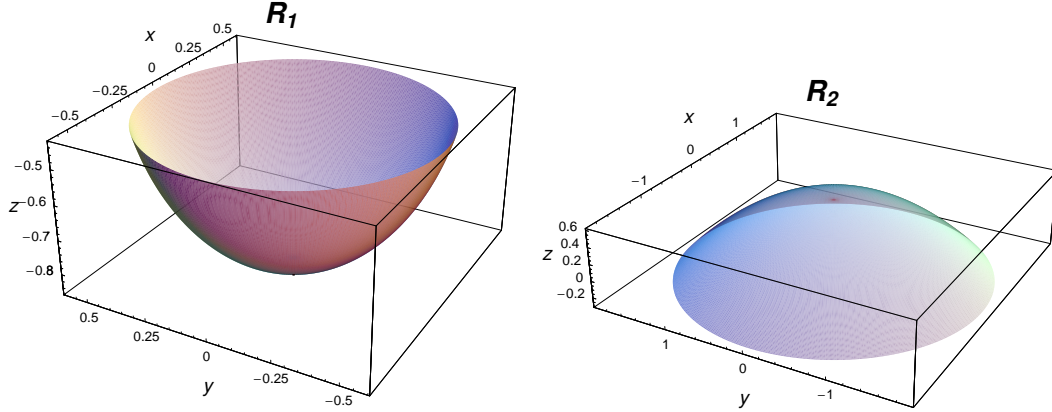


Figure 3: Plots of the reflectors given by (14) (left) and (15) (right) in the explicit data set given in section 4.2.

4.3 Results

Using the data set from subsection 4.2, we conducted a series of numerical tests to establish the validity of the numerical scheme 3.2. Since we have explicit analytical expressions for the two solutions $\rho(m)$ and $z(x)$ describing the two reflectors as given in (14) and (15), respectively, we were able to compute the error of approximation directly. (See also Figure 3 for plots of these reflectors.)

The iterative scheme 3.2 was implemented in MATLAB with the package distmesh [17] for mesh generation in each iteration step and lp_solve [4] for solving the resulting sparse LP. Computations were performed on an Intel I5 Dual Core/2.53GHz with 4GB RAM. Our primary aim is to demonstrate the practicability of the proposed scheme. While there are several ways in which the implementation could be made more efficient, for instance by using a compiled computer language or a more specialized LP solver for transportation problems, we believe that our results show that the proposed scheme is easy to implement and makes it possible to use much finer meshes than a simple discretization scheme.

The mesh generation algorithm employed by the software we used is based on a relaxation scheme of forces in a truss structure [17]. For this, user input is the desired edge length h , not the number of mesh points M (for the domain $\bar{\Omega}$) or N (for the domain \bar{T}), respectively. However, it is not hard to see that the relation between the average edge length h and the number of mesh points M on \bar{D} obeys

$$h \sim \frac{1}{\sqrt{M}}. \quad (16)$$

Indeed, the total area A is proportional to the number of mesh points M times the area of one triangle of a Delaunay triangulation. The area of a triangle is proportional to the square of the average edge length h . The same relation holds of course for the number of mesh points N on \bar{T} . (In Tables 1 and 2 in this section, we show the number of mesh points M, N instead of the edge length h , as we believe the former are more informative.)

We chose a sequence of desired edge lengths $h^{(0)}, h^{(1)}, \dots$. Then, based on the heuristics given above and in section 3.3, we determined the corresponding constraint thresholds with the formula

$$\varepsilon^{(k)} = C \cdot \left(h^{(k)}\right)^a \quad (k = 1, 2, 3, \dots), \quad (17)$$

where C and a are constants. We tested several values for the proportionality constant C and the exponent a . (The heuristic arguments in section 3.3 yields $a = 1$, but we also tested other values.) Some results are summarized in Table 1. As indicated in section 3.3, “large” values for C mean that “many” constraints are included in the LP in each step, which potentially means good approximations, but also high memory usage, and thus we may run out of memory after a few iterations. In contrast, “small” values for C improve memory usage, but may mean that the numerical results may be less accurate approximations of the exact results, or indeed the problem may become unbounded, as typically seems to happen in practice. (See Table 1.) A similar logic holds for the exponent a : For larger a , the threshold $\varepsilon^{(k)}$ decreases faster, leading to less memory usage, but the danger of the problem becoming unbounded.

Table 1 confirms and illustrates these principles. At each iteration, the LP is either unbounded or there is a solution. Interestingly, if there is a solution, it seems largely independent of the number of constraints, as indicated by the fact that the error of approximation varies very little. (See fourth column in Table 1.) This makes some intuitive sense as the most important issue in each iteration is really that we have included the active constraints. As long as all active constraints are included, we get the same solution. If however we did not include any of the constraints for a certain mesh point, the problem becomes unbounded.

In fact, Table 1 illustrates the following feature of the algorithm, which is intuitively quite obvious, although we do not give a formal proof: At each iteration step, there is a critical value $\varepsilon_{\text{crit}}$ such that if $\varepsilon < \varepsilon_{\text{crit}}$, then the LP is unbounded because no constraints involving a certain point are included. If $\varepsilon > \varepsilon_{\text{crit}}$, the LP has a solution. As indicated above, it also appears that if there is a solution, it is likely typically close to the solution of the “full” problem (that is, the problem including all possible constraints). This leads to a possible alternative algorithm for choosing the thresholds $\varepsilon^{(1)}, \varepsilon^{(2)}, \dots$: Instead of choosing each $\varepsilon^{(k)}$ with a pre-determined formula, use the corresponding critical value $\varepsilon_{\text{crit}}$ in each iteration step. This critical value can be found by simple trial and error. While this idea is promising, it is possible that such a scheme may lead to a much slower decay of the maximum error, or even an increase of the error. We did not test this idea in this article, but it would be interesting to analyze it further.

To find out the maximum mesh size on which a solution could be obtained without running out of memory or arriving at an unbounded problem, we “calibrated” the values of C and a . The process is summarized in Table 2.

We obtained best results (that is, largest mesh sizes) for the run with $C = 1.7$ and $a = 1$. Table 3 now summarizes the results for this specific run in more detail. The table shows the errors between the numerical computed approximations and the exact formulas as given in (14) and (15). Figure 4 shows plots of the approximation errors for the reflectors as functions on the input and the output apertures.

These results are a good indication that the scheme converges, and that the error decreases proportional to $M^{-\alpha}$, or equivalently proportional to $h^{2\alpha}$, where roughly $0.5 \lesssim \alpha \lesssim 1$. (The results from Figure 1 appear to indicate $\alpha \simeq 1$, but in other data, we also encountered results where α seemed to be closer to 0.5.) The results also illustrate the practicability of the scheme. Note that the number of mesh points was increased by a factor of about 16 from iteration 1 to iteration 6. In fact the problem in iteration 6 would be impossible to solve with a simple discretization scheme due to the size of the problem. In iteration 6, we were able to reduce the number of constraints used to only about 15% of the number of constraints used for the “naive” simple discretization scheme 3.1.

C	a	sequence of mesh points	notes	max no. constr.	% of possible constr.
1	1.1	284;455;724	unbounded on 724	N/A	N/A
1.2	1.1	284;455;724;1147	unbounded on 1147	N/A	N/A
1.3	1.1	284;455;724;1147	Max err 0.0018575	222220	16.9%
1.5	1.1	284;455;724;1147	Max err 0.0018521	258185	19.6%
2	1.1	284;455;724;1147	Max err 0.0018521	342846	26.1%
5	1.1	284;455;724;1147	Max err 0.0018521	713571	54.0%
0.8	1	284;455;724	unbounded on 724	N/A	N/A
0.9	1	284;455;724;1147	unbounded on 1147	N/A	N/A
1	1	284;455;724;1147	Max err 0.0018521	226231	17.1%
1.5	1	284;455;724;1147	Max err 0.0018521	340038	25.7%
1	1.5	284;455	unbounded on 455	N/A	N/A
2	1.5	284;455	unbounded on 455	N/A	N/A
3	1.5	284;455;724;1147	unbounded on 1147	N/A	N/A
3.5	1.5	284;455;724;1147	unbounded on 1147	N/A	N/A
4	1.5	284;455;724;1147	Max err 0.0018521	224140	16.9%
5	1.5	284;455;724;1147	Max err 0.0018521	282415	21.4%

Table 1: Table summarizing the results for runs of the iterative scheme with different values for the constants C and a given in (17). The third column shows the sequence of mesh points $M^{(k)}$ for \bar{D} . (So for instance, in the first row, in the initial discretization, there were 284 points. In the first iteration, there were 455 mesh points. The second iteration had 724 points, but the problem was unbounded.) The third column notes if the problem became unbounded before reaching 1147 mesh points. If the total run was completed up to 1147 mesh points, the third column gives the maximum norm $\|\rho_{\text{num}} - \rho_{\text{ana}}\|_{\infty}$ of the error between the resulting numerical solution ρ_{num} and the analytic solution ρ_{ana} . The fifth column shows the number of constraints for the case of 1147 mesh points, if the run reached this maximum number. The last column indicates the corresponding percentage of number of constraints relative to the total number of constraints in the “brute force” scheme 3.1. (So for example in the last row, there were only 21.4% of all possible constraints used in formulating the last LP, thus effectively reducing the number of constraints needed by 78.4%.)

C	a	no. iterations	no. mesh points	notes	no. constr.	% of possible constr.
0.8	1	2	724	Max err 0.00148	101174	19.4%
0.8	1	3	1148	Unbounded	178976	13.6%
1	1	3	1148	Max err 0.0018521	226231	17.2%
1.2	1.1	3	1148	Max err 0.0018575	203535	15.5%
1.3	1.1	3	1148	Max err 0.0018575	222220	16.9%
1.5	1.1	3	1148	Max err 0.0018521	258185	19.6%
2	1.1	3	1148	Max err 0.0018521	342846	26.1%
1	1	4	1824	Unbounded	446239	13.5%
1.5	1	4	1824	Max err 0.00131	688163	20.7%
1.6	1	4	1824	Max err 0.00131	733894	22.05%
1.5	1	5	2882	Max err 0.00069	1373976	16.5%
1.6	1	5	2882	Max err 0.00069	1472275	17.7%
1.5	1	6	4536	Unbounded	2650000	12.8%
1.6	1	6	4536	Max err 0.00067	2856627	13.88%
1.7	1	6	4536	Max err 0.00067	3057070	14.86%
1.6	1	7	7130	Out of memory	N/A	N/A
1.7	1	7	7130	Out of memory	N/A	N/A

Table 2: Table illustrating the process of calibrating C and a from (17) for maximum mesh sizes. Each row represents a different run of the iterative numerical scheme. The third column is the number k of iterations in the run and the fourth is the number $M^{(k)}$ of points in the mesh for \bar{D} in the last iteration. The fifth column notes the result of the run – if a solution was found, the maximum error is given. If not, the linear program either became unbounded, or the computer ran out of memory. The last two columns are the number of constraints used for the last iteration and the percentage of total possible constraints (compared to the “brute force” scheme 3.1).

k	h	M	N	Constraints	Max error R_1	L_2 error R_1	Max error R_2	L_2 error R_2
1	0.12	284	278	78,952	0.0048	0.00143	0.008	0.0021
2	0.096	455	450	85942	0.0022	0.00076	0.0047	0.0014
3	0.0768	724	721	183308	0.00148	0.00056	0.0039	0.0012
4	0.06144	1148	1146	381971	0.0012	0.00039	0.00185	0.00044
5	0.049152	1824	1810	779053	0.00060	0.00021	0.0013	0.00033
6	0.0393216	2882	2879	1,568,533	0.00059	0.00019	0.00069	0.00016
7	0.0314573	4536	4525	3,057,070	0.00045	0.00010	0.00067	0.00027

Table 3: Error for a run of the iterative scheme 3.2 with $C = 1.7$ and $a = 1$ in (17). First column: iteration number. Second column: average edge length h . Third and fourth columns: number of mesh points for the domains of reflector 1 and reflector 2, respectively. Fifth column: resulting number of constraints in the LP. The scheme stopped after iteration 7 because there was not enough memory available for the next iteration. The maximum error as a function of total mesh points $N_{\text{tot}} = N + M$ decayed in the form $E_{\text{max}} \sim (N_{\text{tot}})^\alpha$ with $\alpha \approx -0.82$ for reflector 1 and $\alpha \approx -0.95$ for reflector 2. (These values for α were obtained through least squares curve fitting.)

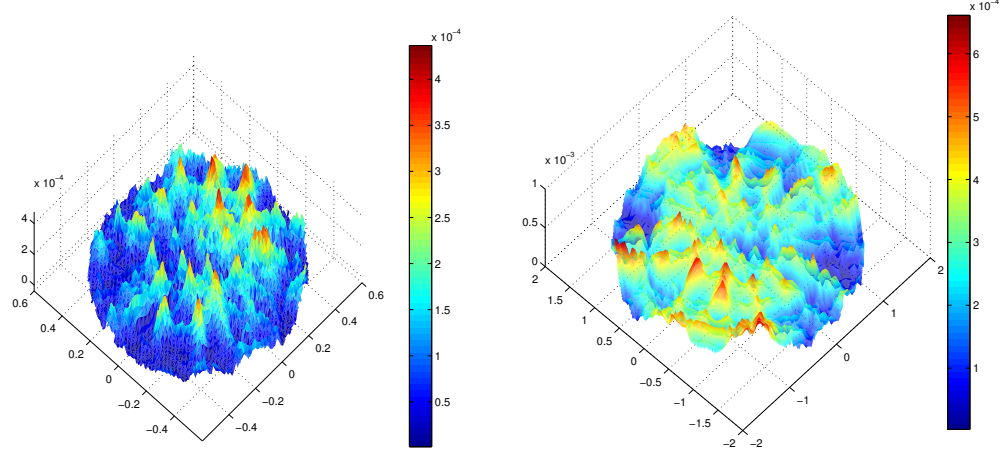


Figure 4: Error surface plots for iteration 7 in Table 3 for reflector 1 (left) and reflector 2 (right). For better visibility, the plots were obtained by interpolating the error at each mesh point to a continuous function. Note that the error distribution appears to be relatively uniform, although the “spikes” are noticeably arranged in triangular patterns and in a fashion somewhat reminiscent of spokes on a wheel. This may be an artifact of an iterated error in conjuncture with triangular meshes.

5 Conclusion and Future Work

We proposed two numerical schemes for solving an infinite dimensional optimal transportation problem arising in reflector design: A straightforward discretization and an improved iterative scheme, which uses knowledge of the previous solution in each step to reduce the number of constraints. This scheme is easily adapted to similar transportation problems arising in beam shaping problems, e.g. in [9] and [11]. We showed that this new scheme is easy to implement and makes it possible to solve the problem on much finer meshes.

There are a number of possible directions for future research. We did not rigorously prove that the scheme converges, although we strongly expect that it does. In fact the decreasing error shown in Table 3 gives evidence for this, at least in our example. It would be valuable to have a rigorous proof of convergence.

Another direction for further investigation is to change the selection algorithm for the threshold values $\varepsilon^{(1)}, \varepsilon^{(2)}, \dots$ as suggested in section 4.3: It is a straightforward conjecture that in each step, there is a critical threshold inclusion number such that the problem is unbounded for values of ε below that number. One way to select a value of ε is to pick a value for $\varepsilon^{(k)}$ close to the critical value $\varepsilon_{\text{crit}}$. It would be interesting to flesh out this ideas and analyze the results more closely.

Another avenue of research is to see whether our proposed solution algorithm can be made parallizable.

It would also be worthwhile to investigate whether the existing PDE-based schemes for solving the transportation cost with quadratic costs [2, 3, 7, 23] can be adapted to the problem at hand.

Acknowledgment

T.G. gratefully acknowledges the contribution of V. Oliker to unpublished collaborative work on an algorithm similar to the one described in this paper for a different reflector design problem. This work was done during 2002 to 2004 at Emory University.

References

- [1] John W. Barrett and Leonid Prigozhin. A mixed formulation of the Monge-Kantorovich equations. *M2AN Math. Model. Numer. Anal.*, 41(6):1041–1060, 2007.
- [2] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numer. Math.*, 84(3):375–393, 2000.
- [3] Jean-David Benamou, Brittany D. Froese, and Adam M. Oberman. Two numerical methods for the elliptic Monge-Ampère equation. *M2AN Math. Model. Numer. Anal.*, 44(4):737–758, 2010.
- [4] Michel Berkelaar, Kjell Eikland, and Peter Notebaert. lp_solve 5.5, open source (mixed-integer) linear programming system. Software, May 1 2004. Available at <http://lpsolve.sourceforge.net/5.5/>.
- [5] Y. Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Comm. in Pure and Applied Math.*, 44, 1991.
- [6] L.A. Caffarelli. Allocation maps with general cost functions. *Partial Differential Equations and Applications*, 177:29–35, 1996.
- [7] B. D. Froese and A. M. Oberman. Fast difference solvers for singular solutions of the elliptic Monge-Ampère equation. *J. Comput. Phys.*, 230:818–834, 2011.
- [8] W. Gangbo and R. J. McCann. Optimal maps in Monge’s mass transport problem. *C. R. Acad. Sci. Paris Sér. I Math.*, 321(12):1653–1658, 1995.
- [9] T. Glimm and V. Oliker. Optical design of single reflector systems and the Monge-Kantorovich mass transfer problem. *J. Math. Sci. (N. Y.)*, 117(3):4096–4108, 2003.
- [10] Tilmann Glimm. A rigorous analysis using optimal transport theory for a two-reflector design problem with a point source. *Inverse Problems*, 26(4):045001, 16, 2010.
- [11] Tilmann Glimm and Vladimir Oliker. Optical design of two-reflector systems, the Monge-Kantorovich mass transfer problem and Fermat’s principle. *Indiana Univ. Math. J.*, 53(5):1255–1277, 2004.
- [12] Tobias Graf. *On the Near-Field Reflector Problem and Optimal Transport*. PhD thesis, Emory University, 2010.
- [13] Eldad Haber, Tauseef Rehman, and Allen Tannenbaum. An efficient numerical method for the solution of the L_2 optimal mass transfer problem. *SIAM J. Sci. Comput.*, 32(1):197–211, 2010.

- [14] V. I. Oliker and L. D. Prussner. On the numerical solution of the equation $(\partial^2 z / \partial x^2)(\partial^2 z / \partial y^2) - ((\partial^2 z / \partial x \partial y))^2 = f$ and its discretizations. I. *Numer. Math.*, 54(3):271–293, 1988.
- [15] V.I. Oliker. Mathematical aspects of design of beam shaping surfaces in geometrical optics. In *Trends in Nonlinear Analysis*, ed. by M. Kirkilionis, S. Krömker, R. Rannacher, F. Tomi, pages 191–222. Springer-Verlag, 2002.
- [16] Vladimir Oliker. Designing freeform lenses for intensity and phase control of coherent light with help from geometry and mass transport. *Archive for Rational Mechanics and Analysis*, 201:1013–1045, 2011. 10.1007/s00205-011-0419-x.
- [17] Per-Olof Persson and Gilbert Strang. A simple mesh generator in Matlab. *SIAM Rev.*, 46(2):329–345 (electronic), 2004.
- [18] Ludger Rüschendorf and Ludger Uckelmann. Numerical and analytical results for the transportation problem of Monge-Kantorovich. *Metrika*, 51(3):245–258 (electronic), 2000.
- [19] Cédric Villani. *Optimal transport: Old and new*, volume 338 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 2009.
- [20] Xu-Jia Wang. On the design of a reflector antenna. *Inverse Problems*, 12(3):351–375, 1996.
- [21] Xu-Jia Wang. On the design of a reflector antenna. II. *Calc. Var. Partial Differential Equations*, 20(3):329–341, 2004.
- [22] Roland Wunderling. *Paralleler und objektorientierter Simplex-Algorithmus*. PhD thesis, Technische Universität Berlin, 1996.
- [23] V. Zheligovsky, O. Podvigina, and U. Frisch. The Monge-Ampère equation: various forms and numerical solution. *J. Comput. Phys.*, 229(13):5043–5061, 2010.